The Stata Journal (2004)
4, Number 4, pp. 421–428

Confidence intervals for the kappa statistic

Michael E. Reichenheim
Instituto de Medicina Social
Universidade do Estado do Rio de Janeiro, Brazil

Abstract. The command `kapci` calculates $100(1-\alpha)\%$ confidence intervals for the kappa statistic using an analytical method in the case of dichotomous variables or bootstrap for more complex situations. For instance, `kapci` allows estimating CI for polychotomous variables using weighted kappa or for cases in which there are more than 2 raters/replications.

Keywords: st0076, kapci, reliability, kappa statistic, confidence intervals

1 Description

`kapci` calculates the confidence interval (CI) for the kappa statistic of interrater agreement using an analytical method in the case of dichotomous variables (Fleiss 1981) or bootstrap for more complex situations (Efron and Tibshirani 1993; Lee and Fung 1993).

Computer efficiency is the main advantage of using an analytical procedure. Alas, to the best of the author's knowledge, no such method has yet been developed to accommodate more complex analysis beyond the simple 2×2 case. Although quite computationally intensive and time consuming, the bootstrap method may be an interesting option to calculate confidence intervals when variables have more than two levels; if three or more raters/replications are involved; and, above all, when a weighted kappa is envisaged. As a compromise between efficiency and necessity, `kapci` implements a specially developed subroutine to handle the analytical solution or Stata's `bs` (see [R] **bstrap**) program for the bootstrap when required.

Details about the bootstrap procedure can be found in Stata's documentation. As for the subroutine, it is based on the asymptotic variance developed by Fleiss, Cohen, and Everitt (1969; Fleiss 1981, equations 13.15–13.18). In `kapci`, calculations use an algebraically modified formulation directly taken from Stata's `sskdlg` (Reichenheim 2000), which is based on Cantor (1996). Here the standard error of kappa needed for the confidence intervals is obtained via

$$Q = (1-\pi_e)^{-1}\left[\sum_i \pi_o\Big\{(1-\pi_e)-(\pi_{.i}+\pi_{i.})(1-\pi_o)\Big\}^2 + (1-\pi_o)^2 \right.$$
$$\left. \sum\sum_{i\neq j} \pi_{ij}(\pi_{.j}+\pi_{j.})^2 - (\pi_o\pi_e - 2\pi_e + \pi_o)^2 \right] \qquad (1)$$

where, given the particular situation of a 2×2 table in which $i = 1$ and $j = 2$, $\pi_e = \pi_{1.}\pi_{.1} + \pi_{2.}\pi_{.2}$ is the expected proportion under the hypothesis of chance agreement, and $\pi_o = \pi_{11} + \pi_{22}$ is the observed proportion, i.e., the sum of the diagonal cells. Since Q equals the variance of kappa times the sample size,

$$\text{s.e.}(\widehat{\kappa}) = \sqrt{Q/N}$$

and an approximate $100(1-\alpha)\%$ confidence interval for κ is

$$\widehat{\kappa} - c_{\alpha/2}\ \text{s.e.}(\widehat{\kappa}) \leq \kappa \leq \widehat{\kappa} + c_{\alpha/2}\ \text{s.e.}(\widehat{\kappa})$$

Equation (1) implies that the standard error of $\widehat{\kappa}$ depends on the point estimate. Therefore, unless $\widehat{\kappa} = 0$, this standard error is usually quite different from that used to calculate the z-statistic in `kap`. For obvious reasons, confidence bounds naively calculated this way are completely misleading.

2 Syntax

`kapci` *varlist* [`if` *exp*] [`in` *range*] [`,` `estim(an bc p n bsall)`
`wgt(w w2` *user_wgt*`)` `reps(`#`)` `size(`#`)` `seed(`#`)` `every(`#`)` `level(`#`)` `tab`
`wide` `saving(`*filename*`)` `replace` `nomsg`]

`by ...:` may be used with `kapci`; see [R] **by**.

3 Options

`estim(`*estimid*`)` requests the type of confidence interval to be displayed. The following *estimid* are available: `an` (analytical), `bc` (bias corrected), `p` (percentile), and `n` (normal). The first is only suitable (it is the default) when the data are dichotomous or two raters/replications are involved. Otherwise, bootstrap is needed, and any of the other three may be chosen. `bc` is the default. All bootstrap estimations may be displayed at once with *estimid* `bsall`.

`wgt(`*wgtid*`)` specifies the type of weight that is to be used to weight disagreements. This option is ignored if there are more than two raters/replications (*varlist* ≥ 3). As in `kap` (see [R] **kappa**), user-defined weights can be created using `kapwgt`. However, `wgt(w)` uses the “prerecorded” weights $1 - |i-j|/(k-1)$, where i and j index the rows and columns of the ratings by the two raters and k is the maximum number of possible ratings. `wgt(w2)` uses the “prerecorded” weights $1 - \{(i-j)/(k-1)\}^2$.

`reps(`#`)` specifies the number of bootstrap replications (B) to be performed. This option is ignored if `estim(an)` is requested or if the data are dichotomous and option `estim()` is omitted. The default number of bootstrap replications has been set to 5 for syntax testing only. In general, `reps()` must be increased when analyzing real data.

`size(#)` specifies the bootstrap size of the samples to be drawn. The default is `_N`, meaning that samples are drawn the same size as the data. The option is ignored under the same conditions as `reps()`.

`seed(#)` specifies the initial value of the random-number seed used by `bs` running under `kapci` when bootstrap is requested or needed. This option is useful for reproducibility of results. # should be specified as an integer. `seed()` is ignored under the same conditions as `reps()`.

`every(#)` specifies that results be written to disk every #th replication. This option should be specified only in conjunction with `saving()` when performing bootstraps that take a very long time. This will allow recovery of partial results should the computer crash. `every()` is ignored under the same conditions as `reps()`.

`level(#)` specifies the confidence level, as a percentage, for the confidence interval. Default is `level(95)`.

`tab` displays all possible two-way tabulations of the assessments.

`wide` requests the display of wide two-way tables. Unless this option is specified, wide tables are broken into pieces to enhance readability. This option is ignored if `tab` is omitted.

`saving(`*filename*`)` dumps B bootstrapped kappas to *filename*`.dta`. If `by ...:` is requested, dumping goes to separate files according to by-groups in the by-variable. As many files (`.dta`) are created as there are by-groups, which are indexed accordingly from 1 to k and respecting the ascending order of values (e.g., *filename*`1.dta`, *filename*`2.dta`, ..., *filenamek*`.dta`).

`replace` indicates that the file specified by `saving()` may exist, and, if it does, it should be overwritten.

`nomsg` suppresses the printing of a warning message that is automatically displayed when `reps()` > 100.

4 Example

To illustrate `kapci`, let's use data relating to a replicated binary variable.

```
. use kapci.example.dta, clear
. kap meas1_G2 meas2_G2
                 Expected
Agreement   Agreement     Kappa   Std. Err.         Z      Prob>Z
-----------------------------------------------------------------
  88.14%      61.25%     0.6938     0.0650      10.67      0.0000
```

```
. kapci meas1_G2 meas2_G2, tab
  meas1_G2 │  meas2_G2 (2 levels)
(2 levels) │          0          1 │     Total
───────────┼───────────────────────┼──────────
         0 │         48         12 │        60
         1 │         16        160 │       176
───────────┼───────────────────────┼──────────
     Total │         64        172 │       236
                                         N=236
────────────────────────────────────────────────
 Kappa (95% CI) = 0.694 (0.589 - 0.799)    (A)
────────────────────────────────────────────────
 A = analytical
```

The kappa point estimates calculated by `kap` and `kapci` are the same, as they should be. Since this analysis relates to a dichotomous variable replicated only once, the default uses the analytical estimation procedure. Nevertheless, bootstrap estimates may also be requested:

```
. kapci meas1_G2 meas2_G2, estim(bsall) reps(1000) seed(1234321)
This may take quite a long time. Please wait ...
                               B=1000  N=236
────────────────────────────────────────────────
 Kappa (95% CI) = 0.694 (0.579 - 0.789)    (BC)
                        (0.580 - 0.789)    (P)
                        (0.588 - 0.800)    (N)
────────────────────────────────────────────────
 BC = bias corrected, P = percentile, N = normal
```

All three types of estimates are quite close to the analytical confidence bounds, especially the normal type (as expected). In any case, the confidence intervals calculated by `kapci` are quite acceptable, since $n = 236$. However, as the following output shows, a very different picture would have been found with a much smaller sample size. Since reliability studies are often of this limited size, the need to account for precision issues becomes quite evident.

```
. set seed 654321
. sample 20
(189 observations deleted)
. kapci meas1_G2 meas2_G2
                                        N=47
────────────────────────────────────────────────
 Kappa (95% CI) = 0.778 (0.576 - 0.981)    (A)
────────────────────────────────────────────────
 A = analytical
```

In this comparison, the 1,000 bootstraps took over half a minute on a 1 GHz PC. Clearly, requesting any of the `bs` options for this simple situation is not the best of choices. The full strength of `kapci`'s bootstrap options only comes to bear in more complicated scenarios. The following output illustrates an intra-observer reliability analysis for a six-level ordinal variable.

```
. kapci meas1 meas2, w(w2) r(1000) sa(external) replace se(12345) nomsg tab wide

  meas1 (6 |                      meas2 (6 levels)
   levels) |        0         1         2         3         4         5 |     Total
-----------+------------------------------------------------------------+----------
         0 |        6         2         2         0         0         0 |        10
         1 |        2        10         4         2         2         0 |        20
         2 |        0         6        16         4         2         2 |        30
         3 |        2         4         6        36         6         4 |        58
         4 |        0         0         2         8        38        10 |        58
         5 |        0         0         2         4         4        50 |        60
-----------+------------------------------------------------------------+----------
     Total |       10        22        32        54        52        66 |       236

                                 B=1000  N=236
------------------------------------------------
 Kappa (95% CI) = 0.790 (0.722 - 0.848)    (BC)
------------------------------------------------
 BC = bias corrected
```

Since the variable involved is polychotomous, bootstrap using the default bias-corrected estimation has been activated. This also enables estimating confidence bounds using intraclass correlation-equivalent quadratic weights (Kraemer 1980). Note that 1,000 B estimates have been saved in `external.dta` for further use.

Assume now that the preceding analysis relates to a clinical condition and that around 28% of those 236 subjects are in a more critical state. One might suspect that measurement reliability is lower for less severe cases precisely because of the subtleties of signs and symptoms in this group. The following output attempts to shed some light on the matter.

```
. by severe: kapci meas1 meas2, w(w2) r(1000) se(12345) sa(severity) nomsg

-------------------------------------------------------------------------------
-> severe = No
                                 B=1000  N=170
------------------------------------------------
 Kappa (95% CI) = 0.751 (0.675 - 0.829)    (BC)
------------------------------------------------
 BC = bias corrected

-------------------------------------------------------------------------------
-> severe = Yes
                                 B=1000  N=66
------------------------------------------------
 Kappa (95% CI) = 0.902 (0.791 - 0.968)    (BC)
------------------------------------------------
 BC = bias corrected
```

On the face of it, the suspicion appears to have been corroborated. According to the point estimates, reliability in each group can be held as quite different. Yet, acknowledging the confidence intervals may call this into question. In fact, further bootstrapping the analysis and formally contrasting the lower 95% confidence bound of the more severely ill with the upper bound of the less ill requires a more conservative perspective.

The ensuing output shows the command sequence used to find out how many times these limits effectively cross. All that is needed is to bootstrap the difference between the required boundary values first obtained (i.e., saved; see the next section) by running kapci on either group and then simply count the instances one limit is above or below the other. The bootstrap routine using kapci underneath with reps() set to 400 is in the output below.

```
. prog list _bs_cross

_bs_cross, rclass:
  1.        syntax [, reps(integer 400)]
  2.        kapci meas1 meas2 if severe==1, wgt(w2) r(‘reps’)
  3.        local k1_lb=‘r(lb_bc)’
  4.        kapci meas1 meas2 if severe==0, wgt(w2) r(‘reps’)
  5.        local k0_ub=‘r(ub_bc)’
  6.        local cross = ((‘k0_ub’)-(‘k1_lb’))
  7.        return scalar k1_lb = ‘k1_lb’
  8.        return scalar k0_ub = ‘k0_ub’
  9.        return scalar cross = ‘cross’
```

Once the program is run, the following bs data are generated:

```
. set seed 1234321

. bootstrap "_bs_cross" r(cross), rep(100) sa(cross) replace nowarn

command:      _bs_cross
statistic:    _bs_1      = r(cross)

Bootstrap statistics                              Number of obs    =       236
                                                  Replications     =       100

------------------------------------------------------------------------------
Variable      |  Reps  Observed      Bias  Std. Err. [95% Conf. Interval]
--------------+---------------------------------------------------------------
        _bs_1 |   100  .0511259 -.0254353  .0899798  -.1274135   .2296652   (N)
              |                                      -.1442649   .1841811   (P)
              |                                      -.0874539   .4160166  (BC)
------------------------------------------------------------------------------

Note:  N   = normal
       P   = percentile
       BC  = bias-corrected

. use cross.dta, clear
(bootstrap: _bs_cross)
```

(*Continued on next page*)

```
. list

        +-----------+
        |     _bs_1 |
        |-----------|
     1. |  .1223744 |
     2. |  .0204346 |
     3. | -.1233235 |
     4. |   .027285 |
     5. |  -.009013 |
        |-----------|
     6. |  .0498347 |
     7. |  .0269275 |
     8. | -.0223853 |
     9. |  .1030313 |
    10. |  .0067131 |
        |-----------|
    11. |  .1310132 |
    12. |  .0025286 |
    13. |  .1119708 |
    14. | -.0457484 |
      (output omitted)
        +-----------+
```

Inspecting the first 14 B iterations, ten `bs` values turn out to be positive, indicating that the upper bound concerning the less ill is beyond the lower bound of the more severely ill. Note that this follows from how the bootstrap routine was specified (see line 6 of the program shown above). This pattern is confirmed in the table below, which is obtained by splitting the `bs` data at 0 (zero). Notably, nearly two-thirds of the comparisons between the confidence boundaries under scrutiny overlap.

```
. gen cross=1

. replace cross=0 if _bs_1<=0
(37 real changes made)

. label var cross "sev_0_ub crossing sev_1_lb"

. label define yesno 1 "Yes" 0 "No"

. label value cross yesno

. tabulate cross

   sev_0_ub |
   crossing |
   sev_1_lb |      Freq.     Percent        Cum.
------------+-----------------------------------
         No |         37       37.00       37.00
        Yes |         63       63.00      100.00
------------+-----------------------------------
      Total |        100      100.00
```

5 Saved Results

`kapci` saves in `r()`:

`r(kappa)`	point-estimate kappa statistic
`r(se)`	standard error
`r(z)`	z-score
`r(N)`	sample size

Available only if analytical estimation is requested

`r(prop_e)`	expected proportion (agreement)
`r(prop_o)`	observed proportion (agreement)
`r(ub_an)`	analytical upper confidence interval
`r(lb_an)`	analytical lower confidence interval

Available only if bootstrap estimations are requested

`r(ub_bc)`	bias-corrected upper confidence interval
`r(lb_bc)`	bias-corrected lower confidence interval
`r(ub_p)`	percentile upper confidence interval
`r(lb_p)`	percentile lower confidence interval
`r(ub_n)`	normal upper confidence interval
`r(lb_n)`	normal lower confidence interval
`r(bias)`	bias
`r(reps)`	number of bootstrap replications (B)
`r(N_bs)`	bootstrap size of the samples to be drawn

6 References

Cantor, A. B. 1996. Sample size calculations for Cohen's k. *Psychological Methods* 1: 150–153.

Efron, B. and R. J. Tibshirani. 1993. *An Introduction to the Bootstrap*. New York: Chapman & Hall.

Fleiss, J. L. 1981. *Statistical Methods for Rates and Proportions*. 2nd ed. New York: Wiley.

Fleiss, J. L., J. Cohen, and B. S. Everitt. 1969. Large sample standard errors for kappa and weighted kappa. *Psychological Bulletin* 72: 323–327.

Kraemer, H. C. 1980. Extension of the kappa coefficient. *Biometrics* 36: 207–216.

Lee, J. and K. P. Fung. 1993. Confidence interval of the kappa coefficient by bootstrap resampling [letter]. *Psychiatry Research* 49: 97–98.

Reichenheim, M. E. 2000. sxd3: Sample size for the kappa-statistic of interrater agreement. *Stata Technical Bulletin* 58: 41–45. In *Stata Technical Bulletin Reprints*, vol. 10, 382–387. College Station, TX: Stata Press.

About the Author

Michael E. Reichenheim (michael@ims.uerj.br), Departamento de Epidemiologia, Programa de Investigação Epidemiológica em Violência Familiar (PIEVF) / Núcleo de Pesquisa das Violências (NUPEVI), Instituto de Medicina Social (*www.ims.uerj.br*), Universidade do Estado do Rio de Janeiro, Brasil